SELF-CONCATENATED TRELLIS CODED MODULATION WITH SELF-ITERATIVE DECODING

S. Benedetto +, D. Divsalar *, G. Montorsi +, F. Pollara *

+ Communications Group, Politecnico di Torino, Italy

* Jet Propulsion Laboratory, California Institute of Technology, USA

Abstract—**Self-concatenated trellis coded modulation with $b(q-1)$ interleavers is a concatenated coded scheme based on only one rate bq/n convolutional code. The b input information sequences and their permuted versions are connected through $b(q-1)$ interleavers to the bq inputs of the convolutional encoder. A subset of the output, including the b information bits and parity bits, is mapped to modulation signals, e.g., MPSK, MQAM. First we obtain an upper bound to the average maximum-likelihood bit error probability of the self-concatenated convolutional coding scheme for binary modulation which also applies to QPSK modulation. Design rules for the single convolutional code, that maximize the interleavers gain and the asymptotic slope of the error probability curves are presented. Asymptotic error performance and design rules for binary modulation are extended to non-binary modulations for the design of self-concatenated trellis coded modulation. A low-complexity self-iterative decoding algorithm for self-concatenated trellis coded modulation is proposed. Examples of self-concatenated coding/decoding are given for binary modulation with $q=2, b=1$, and 8-PSK modulation with $q=2, b=2$. Finally, simulation results for the examples are given for short and long interleavers, using the self-iterative decoder.**

self-concatenation
trellis code
iterative decoding

1. Introduction

Concatenated coding schemes have been studied by Forney [1]. More recent development in concatenated coding schemes are turbo codes proposed by Berrou et al [2]. Turbo codes are *parallel* concatenated convolutional codes (PCCC) using two or more constituent codes.

Trellis coded modulation (TCM) was proposed by Ungerboeck [3] to achieve power and bandwidth efficiency. Turbo (parallel concatenated) trellis coded modulation was proposed in [4], [5] and [6] using the concept of turbo codes and TCM. Later another scheme called serial concatenated TCM was proposed in [7].

These concatenated coding schemes use a suboptimum decoding process based on iterating an " a posteriori probability (APP) algorithm" [8] applied to each constituent code. A soft-input soft-output (SISO) APP module described in [13] was used in this paper to implement a self-iterative decoder. As examples, we will show the results obtained by decoding a rate 1/3 self-concatenated code for binary PSK, and a self-concatenated TCM with 8-PSK to achieve 2 bps/Hz.

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration, and at the Politecnico di Torino. The work was supported in part by NATO under Research Grant CRG 951208 and by Qualcomm, Inc.

For self-concatenated codes, we obtain analytical upper bounds to the performance of a maximum-likelihood (ML) decoding using analytical tools and notations introduced in [10] and [11]. We propose design rules leading to the optimal choice of a high rate convolutional code that maximize the *interleavers gain* [10] and the asymptotic code performance, for both binary and nonbinary modulation.

The basic concept of self-concatenation scheme[1] considered in this paper, shown in Fig. 1, was later independently proposed by Loeliger [9] for the special case of one interleaver and binary modulation. The structure in Fig. 1, which we will refer to as self-concatenated code with $b(q-1)$ interleavers, is a concatenated coded scheme based on one rate bq/n convolutional code, which accepts the b information sequences and their permuted versions through $b(q-1)$ interleavers at its bq inputs. As for turbo codes, the information data is transmitted once. The scheme in Fig. 1 is best suited for the construction of trellis coded modulation, based on one trellis.

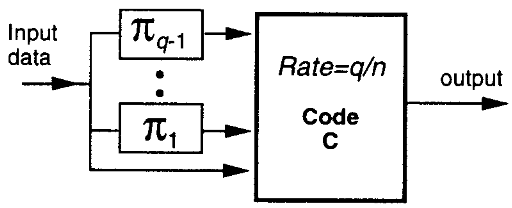

Fig. 1. Self-concatenated code with $b=1$.

In Section 2, we derive analytical upper bounds to the bit error probability of self-concatenated codes, using the concept of "uniform interleavers", i.e., averaging the performance over all possible $q-1$ interleavers. In Section 3, we propose design rules for self-concatenated codes through an asymptotic approximation of the bit error probability bound assuming long interleavers or large signal-to-noise ratios. Section 4 extends the basic concept of self-concatenation, and the asymptotic results on bit error probability for large interleavers to trellis coded modulation schemes. Section 5 describes a self-iterative decoding algorithm for self-concatenated codes for binary and non-binary modulations. Section 6 presents an example of a rate 1/3 self-concatenated code with $q=2$ using a rate 2/4 optimum convolutional code (based on the design rules). Simulation results for this example are given for

[1] Patent pending.

input blocks of 256 and 1024 bits. Finally, an example of construction for a rate 2/3 self-concatenated code from a rate 4/5 optimum convolutional code (based on the design rules for the non-binary case), that uses only a two dimensional 8-PSK for throughput of 2 bps/Hz, is presented. Simulation results using self-iterative decoding for this self-concatenated TCM for input block sizes of 2048 bits and 16384 bits are also given in Section 6.

2. Analytical Bounds on the Performance of Self-Concatenated codes

Consider a linear $(N/R_c, N)$ block code C with code rate R_c, and minimum distance h_m. An upper bound on the bit-error probability (using the union bound) of the block code C over AWGN channels, with coherent detection, and using maximum likelihood decoding, can be obtained as

$$P_b \leq \sum_{h=h_m}^{N/R_c} \sum_{w=1}^{N} \frac{w}{N} A^C_{w,h} Q(\sqrt{2R_c\, h\, E_b/N_0}) \qquad (1)$$

where E_b/N_0 is the signal-to-noise ratio per bit, and $A^C_{w,h}$ represents the number of codewords of the block code C having output weight h, and associated with input sequences of weight w. $A^C_{w,h}$ is the input–output weight coefficient (IOWC). The function $Q(\sqrt{2R_c\, h\, E_b/N_0})$ represents the pairwise error probability which is a monotonic decreasing function of the signal to noise ratio and the output weight h. The Q function is defined as $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt$.

This bound applies to convolutional codes as well if we construct an equivalent block code from the convolutional code. Obviously, this results applies also to concatenated codes including parallel and serial concatenations as well as to the self-concatenated codes discussed in this paper. As soon as we obtain the input–output weight coefficients $A^C_{w,h}$ for a self-concatenated code, we can compute its performance. Clearly the results also applies to independent Rayleigh fading channel if we replace $Q(\sqrt{2R_c\, h\, E_b/N_0})$ by $[\frac{1}{1+R_c\, E_b/N_0}]^h$, where in this case h corresponds to the diversity of the code.

2.1. Self-Concatenated Convolutional Codes

The structure of a self-concatenated convolutional code is composed of $q-1$ interleavers and a single systematic recursive convolutional code C (it will be proved shortly that the code should be recursive) with rate rate $\frac{q}{q+r}$, where r represents the number of parity bits at the output of the encoder for each input symbol. All permuted systematic bits are not transmitted. In this way, the information bit sequence is transmitted once through the channel as for turbo codes, and the code has the equivalent block code representation $(N/R_c, N)$. The self-concatenated code uses $q-1$ independently chosen interleavers each of size N bits, generating a self-concatenated code C with overall rate $R_c = \frac{1}{1+r}$. For illustration only, the structure of a rate 1/2, 4-state self-concatenated convolutional code with $q = 3$, r=1 is shown in Fig. 2.

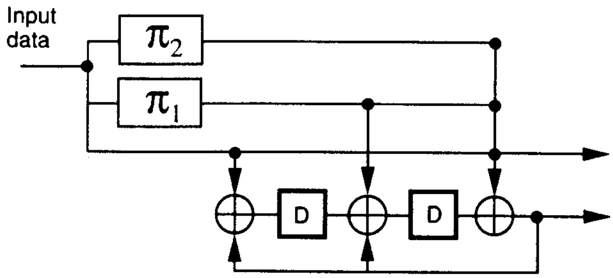

Fig. 2. Example of self-concatenated code, $b = 1$.

2.2. Computation of $A^C_{w,h}$ for Self-Concatenated Codes with Random Interleavers

If the input block N is large, then the computation of $A^C_{w,h}$ for fixed interleavers is an almost impossible task, except for the first few input and output weights. However, the *average* input-output weight coefficients $A^C_{w,h}$ for self-concatenated codes with $q-1$ interleavers can be obtained by averaging (1) over all possible interleavers. This average is obtained by replacing the actual interleavers with abstract interleavers called *uniform interleavers* [10]. A uniform interleaver is defined as a probabilistic device that maps a given input word of weight w into all its distinct $\binom{N}{w}$ permutations with equal probability $p = 1/\binom{N}{w}$, so that the input and output weight is preserved, where N represent the size of the interleaver.

With the knowledge of the IOWC $A^C_{w_1,\ldots,w_q,h}$ for the single convolutional code, using the concept of uniform interleaver, the $A^C_{w,h}$ for the self-concatenated code can be obtained.

According to the properties of uniform interleavers, the ith interleaver transforms an input data of weight w at the input of the self-concatenated code into all its distinct $\binom{N}{w}$ permutations at the ith input of the convolutional code with rate q/n for $i = 1, \ldots, q-1$. As a consequence, each input data block of weight w, through the action of $q-1$ uniform interleavers, is sent to the inputs of the rate q/n convolutional code, generating $\binom{N}{w}^{q-1}$ codewords of the code. Thus, the expression for the IOWC of the self-concatenated code is

$$A^C_{w,h} = \frac{A^C_{w,w,\ldots,w,h}}{[\binom{N}{w}]^{q-1}}, \qquad (2)$$

where $A^C_{w,w,\ldots,w,h}$ is the number of codewords of the convolutional code of weight h associated with the q input words of weight w.

Since we compute the average performance, this means that there will always be, for each value of the signal-to-noise ratio, at least a set of $q-1$ particular interleavers yielding performance better than or equal to that of the $q-1$ uniform interleavers. Using (2) in (1), we can rewrite the upper bound in (1) as

$$P_b(e) \leq \sum_{h=h_m}^{N/R_c} \sum_{w=1}^{N} \frac{A^C_{w,w,\ldots,w,h}}{[\binom{N}{w}]^{q-1}} Q(\sqrt{2R_c\, h\, E_b/N_0}), \quad (3)$$

3. Design of Self-Concatenated Codes

In the following we use analytical tools, definitions, and notations introduced in [10], and [11]. The design of self-concatenated codes is based on the asymptotic behavior of the upper bound in (3) for large interleavers. The reason for the good performance of parallel and serial concatenated codes with input block size of N symbols was that the normalized coefficients $A^C_{w,h}/N$ of a concatenated code decrease with interleaver size for all w, and h. For a given signal-to-noise ratio and large interleavers the maximum component of $A^C_{w,h}/N$ over all input weights w and output weights h, is proportional to N^{α_M}, with corresponding minimum output weight $h(\alpha_M)$.[2] If $\alpha_M < 0$ then for a given SNR the performance of the concatenated code improves as the input block size is increased. If the input block size increases then the size of interleavers used in the concatenated code should also increase. When $\alpha_M < 0$ we say that we have "interleaving gain" [10]. The more negative is α_M the more interleaving gain we can obtain. In order to compute α_M we proceed as follows. Consider a rate $R = q/n$ convolutional code C with memory ν, and its equivalent $(qN/R, qN - q\nu)$ block code whose codewords are all sequences of length qN/R bits of the convolutional code starting from and ending at the zero state. By definition, the codewords of the equivalent block code are concatenations of error events of the convolutional codes. By error event of a convolutional code, we mean a sequence diverging from the zero state at time $t = 0$ and remerging into the zero state at some discrete time $t > 0$. Let $A^C_{w,w,\ldots,w,h,j}$ be the input–output weight coefficients given that the convolutional code generates j error events with q input weights w, and output weight h (see Fig. 3). $A_{w,w,\ldots,w,h,j}$ actually represents the number of sequences of weight h, with q input weights w, and the number of concatenated error events j without any gap between them, starting at the beginning of the block. For N much larger than the memory of the convolutional code, the coefficient $A^C_{w,w,\ldots,w,h}$ of the equivalent block code can be approximated by

$$A^C_{w,w,\ldots,w,h} \sim \sum_{j=1}^{n_M} \binom{N}{j} A^C_{w,w,\ldots,w,h,j} \tag{4}$$

where n_M, the largest number of error events concatenated in a codeword of weight h and generated by q weight-w input sequences, is a function of h and w that depends on the encoder. The large N assumption permits neglecting the length of error events compared to N, which also implies that the number of ways input sequences producing j error events can be arranged in a register of length N is $\binom{N}{j}$. N represents the number of input symbols or, equivalently, trellis steps.

Let us return now to the block code equivalent to the self-concatenated code. In the following, subscript "m" will denote "minimum", and subscript "M" will denote "maximum". Then substituting the above approximation

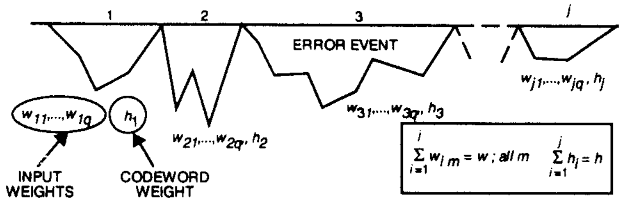

Fig. 3. A code sequence in $A^C_{w,w,\ldots,w,h,j}$

into (3), we obtain the bit error probability bound for the self-concatenated code as

$$P_b(e) \stackrel{\sim}{\leq} \sum_{h=h_m}^{N/R_c} \sum_{w=1}^{N} \sum_{j=1}^{n_M} \frac{w \binom{N}{j}}{N\,[\binom{N}{w}]^{q-1}} A^C_{w,\ldots,w,h,j}\, Q(\sqrt{2R_c h \frac{E_b}{N_0}}) \tag{5}$$

We are interested in large interleaver lengths, and thus use for the binomial coefficient the asymptotic approximation $\binom{N}{j} \sim \frac{N^j}{j!}$. Substitution of this approximation in (5) gives the bit error probability bound in the form

$$P_b(e) \stackrel{\sim}{\leq} \sum_{h=h_m}^{N/R_c} \sum_{w=1}^{N} \sum_{j=1}^{n_M} N^{j-(q-1)w-1} B_{w,h,j}\, Q(\sqrt{2R_c\, h \frac{E_b}{N_0}}) \tag{6}$$

where

$$B_{w,h,j} = \frac{w\,[w!]^{q-1}}{j!} A^C_{w,w,\ldots,w,h,j} \ . \tag{7}$$

Using expression (6), we will obtain some important design rules. The bound (3) to the bit error probability is obtained by adding terms of the first summation with respect to the self-concatenated code weights h. The coefficients of $Q(\cdot)$ in (3) depend, among other parameters, on N. For large N, and for a given h the dominant coefficient of $Q(\cdot)$ is the one for which the exponent of N is maximum. Define this maximum exponent as

$$\alpha(h) \triangleq \max_{w}\{j - (q-1)w - 1\} \ . \tag{8}$$

Evaluating $\alpha(h)$ in general is not possible without specifying the constituent code. Thus, we will consider two important cases, for which general expressions can be found.

3.1. The Exponent of N for the Minimum Weight.

For large values of E_b/N_0, the performance of the self-concatenated code is dominated by the first terms of the summations in h, corresponding to the minimum values $h = h_m$. Using (8), it can be shown that

$$\alpha(h_m) = -q + 1 \tag{9}$$

The RHS is negative for $q \geq 2$, thus there is interleaving gain. The result in (9) shows that the exponent of N corresponding to minimum weight self-concatenated codewords is always negative, thus yielding an interleaver gain at high E_b/N_0. Substitution of the exponent $\alpha(h_m)$ into (6) truncated to the first term of the summation in h yields

[2] α_M is the largest exponent of N as defined in eq. 11.

$$\lim_{\frac{E_b}{N_0}\to\infty} P_b(e) \stackrel{\sim}{\leq} B_m N^{-q+1} Q(\sqrt{2R_c\, h_m\, E_b/N_0}) \quad (10)$$

where the constant B_m is independent of N, and can be computed from (6) and (7).

Expression (10) suggests that, for the values of E_b/N_0 and N where the self-concatenated performance is dominated by the free distance $d_f^C = h_m$, increasing the interleaver length yields a gain in performance. To increase the interleaver gain one should increase the number of interleavers. However, this decreases h_m. To improve the performance with E_b/N_0 one should choose a code such that h_m is large. Therefore there should be an optimum choice for q at high E_b/N_0.

As in serial concatenated codes there are coefficients of $Q(\cdot)$ in h for $h > h_m$, that increase with N. Therefore, we will evaluate the largest exponent of N, defined as

$$\alpha_M \stackrel{\Delta}{=} \max_h\{\alpha(h)\} = \max_{w,h}\{j-(q-1)w-1\}\ . \quad (11)$$

This exponent will allow us to find the dominant contribution to the bit error probability for $N \to \infty$.

3.2. *The Maximum Exponent of N*

We need to treat the cases of non-recursive and recursive encoders separately. For a non-recursive encoder, we can show that $\alpha_M \geq 0$, thus there is no interleaving gain.

For a recursive encoder, after maximization required in (11), we obtain

$$\alpha_M \leq -\left\lfloor\frac{q+1}{2}\right\rfloor \quad (12)$$

Thus there is interleaving gain. So in order to obtain interleaving gain in a self-concatenated code, we should select a recursive encoder.

Next we consider the weight $h(\alpha_M)$ which is the output weight of the code associated to the highest exponent of N.

- For q even, the weight $h(\alpha_M)$ associated to the highest exponent of N is given by

$$h(\alpha_M) = \frac{q d_{f,eff}}{2} + 1 \quad (13)$$

- For q odd, the value of $h(\alpha_M)$ is given by

$$h(\alpha_M) = \frac{(q-3)d_{f,eff}}{2} + h_m^{(3)} + 1\ . \quad (14)$$

Here $h_m^{(3)}$ is the minimum weight of sequences of the code due to the parities generated by weight 3 input sequences. In (13) and (14) $d_{f,eff}$ is the effective free distance of the code, which here is the minimum weight of sequences of the code due to the parities generated by weight 2 input sequences. We can use the Tables of recursive systematic convolutional codes with maximum effective free distances, and maximum $h_m^{(3)}$ which are given in [11] and [12]. However better codes for self-concatenated codes can be found since for rate q/n convolutional codes we have the constraint that the q input sequences entering the q inputs of the q/n recursive convolutional code must have equal weights (ignoring the edge effect due to termination of the code to the all zero state). Examples are given in Section 6 for a rate 1/3 self-concatenated scheme with binary modulation, and input block sizes 256, and 1024 bits.

4. Self-Concatenated Trellis Coded Modulation

We propose a novel method to design self-concatenated TCM, which achieves b bits/sec/Hz, using a single rate $bq/(bq+1)$ recursive systematic binary convolutional encoder where only the $b+1$ outputs of the encoder are mapped to $2^{(b+1)}$ modulation levels. Consider b binary streams entering the self-concatenated TCM. $(q-1)$ interleavers for each input data stream are used. Thus the proposed scheme can be implemented with $b(q-1)$ interleavers. The b input bits plus one parity are mapped to the modulation signal points. In this way, we are using b information bits for every modulation symbol intervals, resulting in b bit/sec/Hz transmission. For illustration, the basic structure of self-concatenated trellis coded modulation for b=3, q=2, and 16QAM modulation that achieves 3 bps/Hz is shown in Fig. 4 (We assume that the bandwidth is $1/T$, where T is the modulation signal symbol duration. This RF bandwidth can be obtained for example by using Nyquist filters with no excess bandwidth).

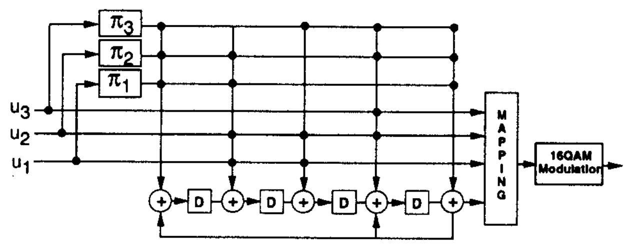

Fig. 4. Example of encoder for self-concatenated trellis coded modulation.

The binary convolutional code and the mapping will be jointly optimized based on maximizing the effective free Euclidean distance of the TCM.

For M-QAM ($M = 2^{2b+2}$) modulations we can also use $b+1$ levels for the I-channel and the next $b+1$ levels for the Q-channel, and achieve a spectral efficiency of $2b$ bits/sec/Hz.

4.1. *Design Criteria for Self-Concatenated TCM*

It can be shown that the dominant term in the transfer function bound on bit error probability of self-concatenated TCM, averaged over all possible interleavers of size N bits, for large N, is proportional to (note that the interleaving gain does not depend on b)

$$N^{-\lfloor(q+1)/2\rfloor} e^{-\delta^2(E_s/4N_o)}$$

where $\lfloor x \rfloor$ represents the integer part of x, and

$$\delta^2 = \frac{q d_{f,\mathrm{eff}}^2}{2}, \tag{15}$$

for q even, and

$$\delta^2 = \frac{(q-3) d_{f,\mathrm{eff}}^2}{2} + (h_m^{(3)})^2, \tag{16}$$

for q odd.

The parameter $d_{f,\mathrm{eff}}$ is the *effective free Euclidean distance* of the TCM which is the minimum Euclidean distance at the output of TCM due to input sequences with Hamming distance 2, with the constraint due to the interleavers discussed before. $h_m^{(3)}$ is the minimum Euclidean distance of TCM sequences generated by input sequences with Hamming distance 3, again with the constraint due to the interleavers, and E_s/N_o is the M-ary symbol signal-to-noise ratio.

Previous results were valid for very large N. On the other hand, for large values of the signal-to-noise ratio E_s/N_o, the performance of self-concatenated TCM is dominated by

$$N^{-q+1} e^{-(h_m^2 E_s/4N_o)}$$

where h_m is the minimum Euclidean distance of the self-concatenated TCM scheme. We considered three different types of mappings for the design of self-concatenated TCM, namely the well known natural mapping, Gray code mapping, and reordered mapping. Examples are given in Section 6 for 2 bps/Hz with 8PSK modulation for input blocks of 2048, and 16384 bits.

5. Iterative Decoding of Self-Concatenated Codes for Binary and Non-Binary Modulations

In previous sections, we have shown analytical results for the performance of self-concatenated codes, when decoded using a ML algorithm. In practice, however, ML decoding of these codes with large N is an almost impossible task. Thus, to acquire practical significance, the above described codes and analytical bounds need to be accompanied by a decoding algorithm of the same order of complexity as the decoder for a single code, yet retaining the performance advantage. In this section, we present an iterative decoding algorithm for self-concatenated codes, with complexity not significantly higher than that needed to decode the single code used.

For decoding of the received sequence, we will use the soft input soft output SISO APP module described in [13]. A functional diagram of the iterative decoding algorithm for self-concatenated codes is presented in Fig. 5, for q=3, b=1.

We will explain how the algorithm works, according to the blocks of Fig. 5 (This iterative decoder can be used for example for the self-concatenated structure in Fig. 2). The blocks labeled "SISO" have two inputs and two outputs. The input labeled $\lambda(c; I)$ represents the reliability of the unconstrained output symbols of the encoder, while

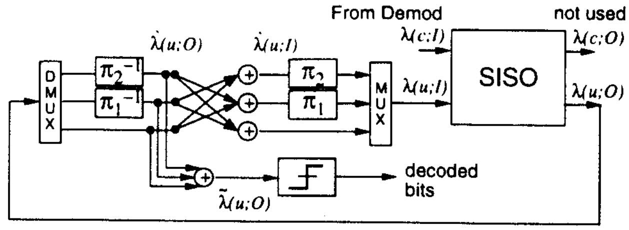

Fig. 5. Iterative decoding algorithm for self-concatenated convolutional code, $q = 3$

that labeled $\lambda(u; I)$ represents the reliability of the unconstrained input symbols of the encoder. Similarly, the outputs represent the same quantities conditioned to the code constraint as they are evaluated by the APP decoding algorithm in the log domain. The SISO APP module updates both reliability of the input and output symbols based on the code constraints. Both outputs of SISO, i.e., $\lambda(c; O)$, and $\lambda(u; O)$ directly generate the "extrinsic" information required for iterative decoding. So there is no need to subtract the the unconstrained input reliability from the output reliability generated by the APP algorithm.

During the first iteration of the self-iterative algorithm: The block "SISO" is fed with the demodulator soft output, consisting of the reliability of symbols received from the channel, i.e., the received output symbols of the encoder. The received reliabilities are processed by the SISO module that computes the extrinsic information of the inputs symbols conditioned on the code constraints. This information is passed through the inverse interleavers (blocks labeled "π_i^{-1}", $i = 1, .., q-1$). Deinterleaved extrinsics after exchange of information are passed through interleavers (blocks labeled "π_i", $i = 1, .., q-1$). The outputs of the interleavers correspond to the reliabilities of the input symbols of the same and only code (self-iteration), and they are sent to the SISO module's port, which corresponds to input symbols, and so on for each iteration.

The reliability of input symbols of the SISO module and the extrinsics for the input symbols of the SISO will be used in the final iteration to recover the information bits.

5.1. *Bit-by-Bit Iterative Decoding using the APP SISO Algorithm in Log Domain.*

For completeness we briefly describe the SISO algorithm based on the trellis section shown in Fig. 6, for a generic code C with input symbol $\mathbf{u}$ and output symbol $\mathbf{c}$. A detailed description is provided in [13]. Consider a code with q input bits and p symbols binary $\{0, 1\}$ or nonbinary. Let the input symbol to the convolutional code $\mathbf{u}_k(e)$ represent $u_{k,i}(e)$; $i = 1, 2, \cdots, q$ the input bits $\{0, 1\}$ on a trellis edge at time k, and let the output symbol of the convolutional code $\mathbf{c}_k(e)$ represent $c_{k,i}(e)$; $i = 1, 2, \cdots, p$ symbols binary or nonbinary.

Define the reliability of a bit Z taking values $\{0, 1\}$ at time k as

$$\lambda_k[Z; \cdots] \triangleq \log \frac{P_k[Z = 1; \cdot]}{P_k[Z = 0; \cdot]}$$

The second argument in the brackets, shown as a dot, may

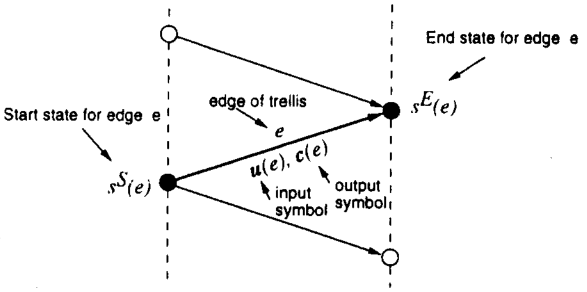

Fig. 6. Trellis Section for the Code C.

represent I, the input, or O, the output, to the SISO. We use the following identity

$$a = \log[\sum_{i=1}^{L} e^{a_i}] \triangleq \max_i{}^*\{a_i\}$$

which can be computed using a look-up table. We define the "max*" operation as a maximization (compare-select) plus a correction term (lookup table). We can replace "max* " with "max" for a small penalty in performance.

The received samples $\{y_{k,i}\}$ at the output of the receiver matched filter are $y_{k,i} = Ax(c_{k,i}) + n_{k,i}$, the noise has variance σ^2 per dimension, and $E\{|x(c_{k,i})|^2\}$=1, which is the assumed channel model ($\frac{E_s}{N_o} = \frac{A^2}{2\sigma^2}$). For binary modulation $x(c_{k,i}) = (2c_{k,i} - 1)$. Without loss of generality, assume the encoder starts (at the beginning of the block) at the all zero state and ends into the all zero state (at the end of block, when termination is used). For an encoder with memory ν, let s represent the state of the encoder, where $s \in \{0, \ldots, 2^\nu - 1\}$.

5.1.1. The APP SISO Algorithm for the Code C.

The *forward* and the *backward* recursions are:

$$\begin{aligned}\alpha_k(s) \quad = \quad & \max_{e:s^E(e)=s}{}^* \{\alpha_{k-1}[s^S(e)] + \sum_{i=1}^{q} u_{k,i}(e)\lambda_k[U_{k,i}; I] \\ & + \sum_{i=1}^{p} \lambda_k[c_{k,i}(e); I]\} + h_{\alpha_k}\end{aligned}$$

$$\begin{aligned}\beta_k(s) \quad = \quad & \max_{e:s^S(e)=s}{}^* \{\beta_{k+1}[s^E(e)] \\ & + \sum_{i=1}^{q} u_{k+1,i}(e)\lambda_{k+1}[U_{k+1,i}; I] \\ & + \sum_{i=1}^{p} \lambda_{k+1}[c_{k+1,i}(e); I]\} + h_{\beta_k}\end{aligned}$$

with initial values, $\alpha_0(s) = 0$, if $s = 0$ (initial zero state) and $\alpha_0(s) = -\infty$, if $s \neq 0$. Similarly, $\beta_n(s) = 0$, if $s = 0$ (final zero state) and $\beta_n(s) = -\infty$, if $s \neq 0$. Recursions are done for $k = 1, \ldots, n-1$, where n represents the total number of trellis steps for the encoder, and h_{α_k}, h_{β_k} are normalization constants. Based on the channel model described above, we have

$$\lambda_k[c_{k,i}(e); I] = -\frac{|y_{k,i} - Ax(c_{k,i}(e))|^2}{2\sigma^2} \qquad (17)$$

If we replace "max* " with "max", we have a Viterbi-type algorithm in the forward and backward directions. The *extrinsic bit information* for $U_{k,j}$; $j = 1, 2 \cdots, q$; can be obtained from:

$$\begin{aligned}\lambda_k(U_{k,j}; O) \quad = \quad & \max_{e:u_{k,j}(e)=1}{}^* \{\alpha_{k-1}[s^S(e)] \\ & + \sum_{\substack{i=1 \\ i\neq j}}^{q} u_{k,i}(e)\lambda_k[U_{k,i}; I] + \sum_{i=1}^{p} \lambda_k[c_{k,i}(e); I] + \beta_k[s^E(e)]\} \\ & - \max_{e:u_{k,j}(e)=0}{}^* \{\alpha_{k-1}[s^S(e)] + \sum_{\substack{i=1 \\ i\neq j}}^{q} u_{k,i}(e)\lambda_k[U_{k,i}; I] \\ & + \sum_{i=1}^{p} \lambda_k[c_{k,i}(e); I] + \beta_k[s^E(e)]\}\end{aligned}$$

At the first iteration all input reliabilities $\lambda_k[U_{k,i}; I]$ are zero. The SISO computes the extrinsic information $\lambda_k(U_{k,j}; O)$ from the above equations, and provides them to itself after exchange of information. The exchange of information for computing the input reliabilities for the next use of SISO can be done as follows

$$\lambda'_k(U_{k,j}; I) = \sum_{\substack{i=1 \\ i\neq j}}^{q} \lambda'_k(U_{k,i}; O)$$

where $\lambda'_k(U_{k,j}; I)$, and $\lambda'_k(U_{k,i}; O)$ represent unpermuted versions of $\lambda_k(U_{k,j}; I)$, and $\lambda_k(U_{k,i}; O)$ respectively.

The self-iterative decoder makes decisions on

$$\tilde{\lambda}_k = \sum_{i=1}^{q} \lambda'_k(U_k; O)$$

by passing it through a hard limiter.

6. Examples and Simulation Results for Self-Concatenated Codes

6.1. Example and Simulation Results for Rate 1/3 Self-Concatenated Code with Binary Modulation

Consider a rate 1/3 self-concatenated code with q=2, and a 16-state recursive convolutional code, (obtained from a table for 2/4 codes with maximum effective free distance in [12]) as shown in the Fig. 7. The simulation results for this code using the iterative decoder in Fig. 8 are shown in Fig. 9(a) for input block N = 256, and(b) for 1024 bits. Two uses of SISO per input block were counted as one iteration.

6.2. Examples and Simulation Results for Self-Concatenated Trellis Coded Modulation

Consider self-concatenated TCM with q=2, constructed from a rate 4/5, 16-state recursive convolutional code, (obtained from a limited search satisfying the proposed criteria for design of self-concatenated TCM). Two examples are considered. The first example uses the structure in Fig. 11 with Gray code mapping, 8PSK modulation, and input block of 2048 bits. Two interleavers, each with size 1024

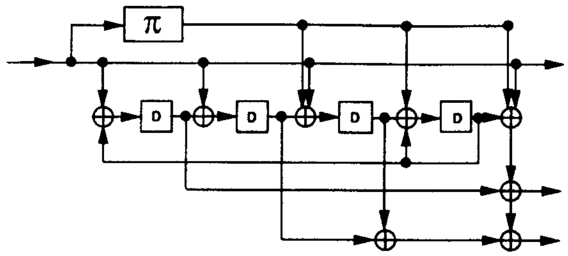

Fig. 7. A rate 1/3, 16-state Self-Concatenated Code with q=2

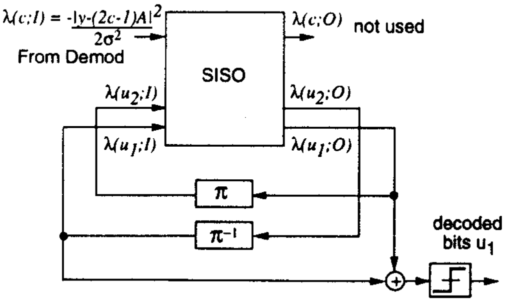

Fig. 8. A self-iterative decoder for q = 2.

were used. The simulation results for the first example using iterative decoding in Fig. 12 is shown in Fig. 13(a). For the second example the structure of self-concatenated TCM is shown in Fig. 10 which uses reordered mapping, and input block of 16384 bits. Two interleavers each with size 8192 bits were used. The simulation results for the second example are shown in Fig. 13(b). For the simulations, the iterative decoder in Fig. 12 was used. Again, two uses of SISO per input block were counted as one iteration.

References

[1] G.D. Forney Jr., *Concatenated Codes*, MIT, Cambridge, MA, 1966.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes", in *Proceedings of ICC'93*, Geneve, Switzerland, May 1993.

[3] G. Ungerboeck, "Channel coding with multilevel phase signaling," *IEEE Trans. Inf. Th.*, Vol.25, pp. 55-67, Jan. 1982.

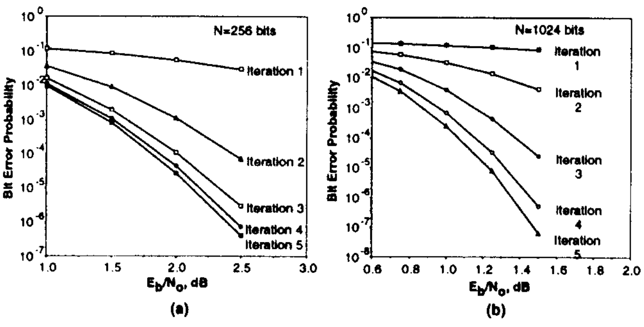

Fig. 9. Simulation results for rate 1/3, 16-state Self-Concatenated Code with q=2, (a) N=256, and (b) N=1024 bits

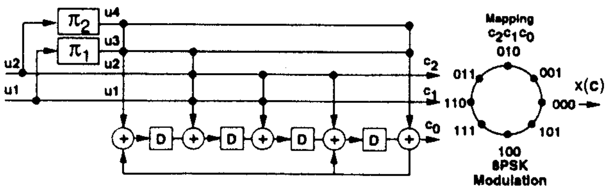

Fig. 10. Self-concatenated trellis coded modulation with 8PSK.

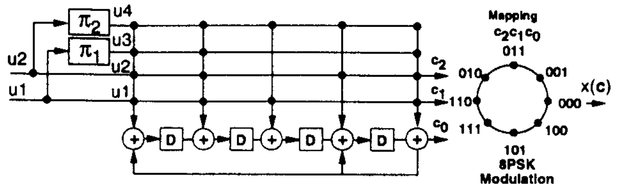

Fig. 11. A Self-Concatenated TCM with Gray code mapping for 8PSK, q=2, b=2 bps/Hz

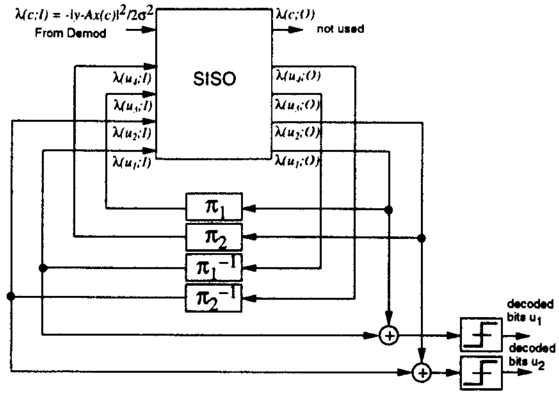

Fig. 12. A self-iterative decoder for self-concatenated TCM q = 2, b = 2 bps/Hz.

[4] S. LeGoff, A. Glavieux, and C. Berrou, "Turbo Codes and High Spectral Efficiency Modulation", *Proceedings of IEEE ICC'94*, May 1-5, 1994, New Orleans, LA.

[5] P. Robertson, and T. Woerz, "Novel Coded modulation scheme employing turbo codes," Electronics Letters, 31st Aug. 1995, Vol. 31, No. 18.

[6] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "Bandwidth Efficient Parallel Concatenated Coding Schemes," IEE Electronics Letters, November 1995, Vol. 31, No. 24.

[7] S. Benedetto, D. Divsalar, G. Montorsi, F. Pollara, "Serial Concatenated Trellis Coded Modulation with Iterative Decoding:Design and Performance". CTMC97

[8] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate". *IEEE Transactions on Information Theory*, pages 284–287, March 1974.

[9] H. A. Loeliger " New turbo like codes," in proceedings of ISIT 97, June 28- July 4 1997, Ulm, Germany.

[10] S. Benedetto and G. Montorsi, "Unveiling turbo-codes: some results on parallel concatenated coding schemes", *IEEE Transactions on Information Theory*, vol. 43, no. 2, Mar. 1996.

[11] S. Benedetto and G. Montorsi, "Design of Parallel Concatenated Convolutional Codes", *IEEE Transactions on Communications*, vol. 44, no. 5, pp. 591–600, May 1996.

[12] D. Divsalar and R. J. McEliece, "The effective free distance of turbo codes", *IEE Electronic Letters*, vol. 32, no. 5, Feb. 29, 1996.

[13] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft input soft output MAP module to decode parallel and serial concatenated codes", *TDA Progress Report 42-127, July–September 1996*, JPL, Pasadena, CA, November 15, 1996.

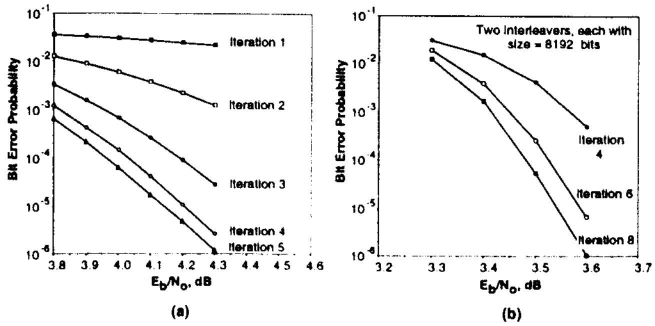

Fig. 13. Simulation results for Self-Concatenated TCM for 8PSK, q=2, (a) N=2048 bits with Gray code mapping, (b) N=16384 bits with reordered mapping